

LANGAGES DE PROGRAMMATION

1. Langage machine

Un processeur exécute séquentiellement des instructions codées sur un octet ou un groupe d'octets. Une suite de valeurs directement exécutables par un processeur constitue un programme en langage machine. Ces instructions et leur codage sont spécifiques à chaque processeur, il est donc pénible et peu productif d'écrire un programme directement en langage machine.

2. Assembleur

On appelle assembleur ou plus exactement langage d'assemblage le fait de coder chaque instruction en langage machine par un code plus explicite. Exemple :

0x6A 0x14 en langage machine x86 s'écrit **push 0x14** en assembleur.

L'assembleur reste assez largement spécifique à chaque processeur et nécessite l'écriture de beaucoup de lignes même pour des programmes simples. Il ne contient pas non plus de mécanisme permettant d'éviter ou de corriger les erreurs. Il est donc encore très peu productif mais il est encore utilisé quand on a besoin de maîtriser précisément le comportement du processeur. C'est un langage de très bas niveau.

3. Langages compilés

On dit qu'un langage de programmation est compilé si le code que l'on écrit est transformé en code machine (compilation) avant d'être exécuté.

Les langages compilés sont en général d'assez bas niveau et ils sont de ce fait relativement rapides.

Exemple de langage compilés :

C, C++, Go, Pascal...

4. Langages interprétés

Un langage de programmation est interprété si la machine exécute un programme spécifique (interpréteur) qui lui va interpréter le code écrit.

Les langages interprétés sont plus lents que les langages compilés mais ils permettent en général une écriture assez concise des programmes.

Exemples de langages interprétés :

Javascript, Php, Bash...

Remarque : Certains langages sont intermédiaires, ils nécessitent la transformation du code écrit en un code qui ne sera pas directement du code machine mais sera interprété efficacement. Java et Python sont dans ce cas.